AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (previously presented) A method of operation within a data processing system that includes a plurality of processing nodes each having access to a set of shared resources, the method comprising:

detecting a failed node within the plurality of processing nodes;

granting access, without delay, to the shared resources that were not subject to access control by the failed node and, at the time the failed node was detected, were not subject to exclusive access by the failed node;

releasing locks to shared resources that were held by the failed node;

remastering, to non-failed nodes, shared resources that were mastered by the failed node;

granting access, to shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node;

performing redo operations of the failed node; and

granting access, after performing redo operations, to all shared resources.

2. (previously presented) The method of claim 1 further comprising:

determining whether the failed node was responsible for controlling access to a first resource; and

upon determining that the failed node was not responsible for controlling access to the first resource,

determining whether, at the time the failure was detected, the failed node had exclusive access to the first resource.

3. (original) The method of claim 2 wherein determining whether the failed node was responsible for controlling access to the first resource comprises inspecting a data structure that indicates, for each shared resource within the set of shared resources, which of the plurality of processing nodes is responsible for controlling access to the shared resource.

4. (original) The method of claim 3 wherein determining whether the failed node was responsible for controlling access to the first resource comprises identifying a data element within the data structure that includes a first component that identifies the first resource and a second component that identifies a processing node responsible for controlling access to the first resource.

5-9. (cancelled)

10. (previously presented) The method of claim 1, wherein remastering further comprises generating a data structure within a first non-failed node that indicates whether a processing node of the plurality of processing nodes, other than the failed node, had access to the shared resources that were mastered by the failed node when the failed node was detected.

11. (cancelled)

12. (previously presented) The method of claim 1, wherein releasing locks further comprises adding an identifier of the shared resources held by the failed node to a validation data structure if the failed node was not responsible for controlling access to the shared resources held by the failed node but had exclusive access to the shared resources held by the failed node when the failure was detected.

13-20. (cancelled)

21. (previously presented) A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to:

detect a failed node within the plurality of processing nodes;

grant access, without delay, to the shared resources that were not subject to access control by the failed node and, at the time the failed node was detected, were not subject to exclusive access by the failed node;

release locks to shared resources that were held by the failed node;

remaster, to non-failed nodes, shared resources that were mastered by the failed node;

grant access, to shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node;

perform redo operations of the failed node; and

grant access, after performing redo operations, to all shared resources.

22. (cancelled)

23. (previously presented) A system comprising:

a data storage device having a set of shared resources stored therein; and

a plurality of processing nodes each having a processing entity and a memory coupled to the processing entity, the memory having program code stored therein which, when executed by said processing entity, causes said processing entity to:

detect a failed node within the plurality of processing nodes;

grant access, without delay, to the shared resources that were not subject to access control by the failed node and, at the time the failed node was detected, were not subject to exclusive access by the failed node;

release locks to shared resources that were held by the failed node;

remaster, to non-failed nodes, shared resources that were mastered by the failed node;

grant access, to shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node;

perform redo operations of the failed node; and

grant access, after performing redo operations, to all shared resources.

24. (cancelled)

25. (previously presented) The method of claim 1, wherein detecting a failed node further comprises determining whether (i) periodic transmissions by the failed node has ceased, (ii) the failed node is non-responsive to communications, or (iii) affirmative failure notification by the failed node.

26. (previously presented) The method of claim 1, further comprising: before completion of the remastering, granting access to at least some of the shared resources that the failed node

controlled and, at the time the failed node was detected, were subject to access by a non-failed node.

27. (previously presented) The method of claim 1, wherein remastering further comprises granting access of the shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node, incrementally as redistribution of access control of shared resources by the failed node to non-failed nodes is performed.

28. (previously presented) The method of claim 27, wherein redistribution of access control of shared resources by the failed node to non-failed nodes further comprises master reassignment and lock list recovery, that identifies locks held to re-mastered resources and to notify the new master of the lock.

29. (previously presented) The method of claim 1, wherein releasing locks further comprises each processing node maintaining a lock list comprising lock data values that correspond to resource locks granted by the node but not released.

30. (previously presented) The method of claim 29, wherein lock data values comprise a resource identifier to identify the resource to which the lock corresponds, a lock holder value that identifies the processing node to which the lock has been granted, and a lock mode value that indicates whether the lock is an exclusive lock or shared lock.

31-41. (Cancelled)

42. (New) The computer-readable storage medium of Claim 21, further comprising:

determining whether the failed node was responsible for controlling access to a first resource; and

upon determining that the failed node was not responsible for controlling access to the first resource,

determining whether, at the time the failure was detected, the failed node had exclusive access to the first resource..

43. (New) The computer-readable storage medium of Claim 42, wherein determining whether the failed node was responsible for controlling access to the first resource comprises inspecting a data structure that indicates, for each shared resource within the set of shared resources, which of the plurality of processing nodes is responsible for controlling access to the shared resource.

44. (New) The computer-readable storage medium of Claim 43, wherein determining whether the failed node was responsible for controlling access to the first resource comprises identifying a data element within the data structure that includes a first component that identifies the first resource and a second component that identifies a processing node responsible for controlling access to the first resource.

45. (New) The computer-readable storage medium of Claim 21, wherein remastering further comprises generating a data structure within a first non-failed node that indicates whether a processing node of the plurality of processing nodes, other than the failed node, had access to the shared resources that were mastered by the failed node when the failed node was detected.

46. (New) The computer-readable storage medium of Claim 21, wherein releasing locks further comprises adding an identifier of the shared resources held by the failed node to a validation data structure if the failed node was not responsible for controlling access to the shared resources held by the failed node but had exclusive access to the shared resources held by the failed node when the failure was detected.

47. (New) The computer-readable storage medium of Claim 21, wherein detecting a failed node further comprises determining whether (i) periodic transmissions by the failed node has

ceased, (ii) the failed node is non-responsive to communications, or (iii) affirmative failure notification by the failed node.

48. (New) The computer-readable storage medium of Claim 21, further comprising: before completion of the remastering, granting access to at least some of the shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node.

49. (New) The computer-readable storage medium of Claim 21, wherein remastering further comprises granting access of the shared resources that the failed node controlled and, at the time the failed node was detected, were subject to access by a non-failed node, incrementally as redistribution of access control of shared resources by the failed node to non-failed nodes is performed.

50. (New) The computer-readable storage medium of Claim 49, wherein redistribution of access control of shared resources by the failed node to non-failed nodes further comprises master reassignment and lock list recovery, that identifies locks held to re-mastered resources and to notify the new master of the lock.

51. (New) The computer-readable storage medium of Claim 21, wherein releasing locks further comprises each processing node maintaining a lock list comprising lock data values that correspond to resource locks granted by the node but not released.

52. (New) The computer-readable storage medium of Claim 51, wherein lock data values comprise a resource identifier to identify the resource to which the lock corresponds, a lock holder value that identifies the processing node to which the lock has been granted, and a lock mode value that indicates whether the lock is an exclusive lock or shared lock.